The Right Things for the Right Reasons
Lessons Learned Achieving CMM Level 5

Patrick W. Cosgriff
Ogden Air Logistics Center, Software Engineering Division

This article discusses lessons we learned on our journey to implementing Capability Maturity Model (CMM) Level 5 processes at the Technology & Industrial Support Directorate (TIS), Software Engineering Division at Hill Air Force Base, Utah. Although ours is not the only way to implement the CMM, we have learned that the only way to use the CMM effectively is to do the right things for the right reasons, using the CMM as a guide. Do not limit solutions to the activities and subpractices listed in the CMM, which are general examples; specific situations require specific solutions. This article describes some of the unique processes we implemented as we strived to understand the underlying CMM principles and looked for solutions that make sense with respect to those principles.

As our organization prepared for our Level 5 assessment, there was no great fear of failure. We by no means thought we had it made, but we were confident we were doing the right things for the right reasons. We had identified areas of risk, most of which involved what the CMM calls "alternate practices." These are methods of satisfying the intent of the CMM that are different than the practices and subpractices listed in the CMM. We were unsure how these alternate practices would be received by an assessment team; however, we decided to do process improvement our way because it made sense and it worked. After all, our ultimate goal was and is to continually improve our performance, not to get a rating.

During the assessment, we were pleased that the highly experienced assessment team agreed that our alternate practices satisfied the intent of the CMM. This article discusses some important processes we have implemented at TIS. All are used because they work well and they make sense—the right things for the right reasons.

Management Sponsorship

You have probably heard again and again that senior management support is crucial to implementing the CMM—if something is not a priority to the people who control an organization's future, it will not be a priority to the organization's members. Management support alone is not enough, but without it, you cannot even get the troops to start.

We were fortunate to have solid management support at all levels, which was born from a desire to improve rather than a desire to win the "get a level" game. We would have done things differently if our only goal was to satisfy the CMM. Our strategic goals are to improve the quality of our products, improve our estimating accuracy, improve cost performance, improve schedule performance, and decrease cycle time. These goals are set quantitatively for the entire division by senior management. Product-line goals are kept consistent with division goals.

Packaging Project Plans

There are approximately 24 different plans referred or alluded to in the CMM. We did not believe this meant we should create 24 different planning documents. We found that when too many documents were required, people began to substitute documents for plans—they spent too much time developing documents that neither represented actual planning activities nor used to manage the projects. We learned to emphasize true planning activities more than documents. Our policy was changed to emphasize that projects must be planned, and the planning must be documented. It is a subtle difference, but by focusing on the activity more than the packaging, we put the emphasis on the important part.

Our current simplified view of a project plan answers five questions:

- What will we accomplish?
- How will we accomplish it?
- How much will it cost?
- How long will it take?
- How much deviation is acceptable?

Requirements must be documented and configured. The project must have processes, procedures, and the necessary support plans. The budget must be agreed to and configured. The project must have a schedule that is baselined and configured. Thresholds or control limits must be established based on the project's capability baseline.

These artifacts are in the form and location most convenient for the people who need to use them. A big binder full of material may be convenient for an assessment team, but we are in the business of developing and maintaining software—not in the business of being assessed. Usability is the key. A plan that is unusable becomes shelfware. Our organizational policy defines project-planning requirements, but the Extended SEPG (ESEPG) (see "Product Lines" section) individually decide the best way to package processes and plans.

Our product-line approach also simplifies project planning. By grouping similar projects together into product lines, many activities, such as configuration management, quality assurance, intergroup coordination, and process documentation, were standardized. This greatly reduced the planning necessary for individual projects.

Operational Definitions

What is the difference between a process and a procedure? What is a defect? Having terms consistently defined across your organization is more important than the correctness of the definition. Operational definitions are critical, and the issue becomes more critical the larger the organization. Defining terminology is crucial to communicating ideas consistently throughout the organization. We needed to define terms such as process, procedure, software development plan, defect, requirements, capability baseline, threshold, and control limit. Before standardizing, even the term "project" had a different meaning from product line to product line.

Software Quality Assurance and Configuration Management

According to the software CMM, Version 1.1, "The purpose of Software Quality Assurance [SQA] is to provide management with appropriate visibility into the process being used by the software project and of the products being built." Intuitively, it would seem to make sense to lump both product and process quality assurance into one KPA; however, when we tried to implement this, it was clear that product quality assurance and process quality assurance were different activities. They require different skills, different training, and are done at different times. The assessment team considered our implementation of SQA to be an alternate practice.

The focus of our alternate practice was to break process quality assurance (QA) apart from product QA. We formed a team called the Quality Engineering Support Team (QuEST) (see "Software Quality Assurance in a CMM Level 5 Organization," page 11), whose primary responsibility was to perform process QA. The product QA duties were performed by our configuration management (CM) group. Both QuEST and CM are independent groups outside of the chain of command of the projects. This is essential to ensure cost or schedule problems do not cause quality to suffer. Their only responsibility is to assure quality products are being produced and quality processes are being followed. Project cost and schedule performance is not their concern.

Product Quality Assurance

To understand how the CM group performs the product QA role for us, you must understand how the CM group operates in our environment. Many people still think of CM as happening only at the beginning and at the end of a project. At OO-ALC/TIS, configuration specialists are involved throughout the process. They check in and check out baseline work products and developmental work products. They schedule and attend peer reviews, where they often act as recorder, taking minutes and recording anomalies. They ensure that the appropriate people have signed off on any anomalies that require further investigation or rework.

They are thought of as people like traffic officers who control the flow of work products between groups and phases in the project. In preparation for peer reviews, they review the work products for compliance to style guides, templates, checklists, etc. This leaves the technical people free to focus on the technical accuracy of the work product being reviewed instead of style-guide details.

Process Quality Assurance

Process QA is handled by our QuEST. QuEST is a three-person team that audits projects for compliance to division policy. They audit about 80 percent of the division every six to eight months. Originally, QuEST only reviewed software engineering projects. Now, they also audit non-software engineering projects and support groups. They also review the management chain of each branch, as well as the Executive Board, division chief, and division staff as a group, for compliance to division policy.

QuEST audits projects with a set of questions that are taken directly from our organizational policy. QuEST audits are performed by interviewing project personnel, including practitioners, lead engineers, and project managers. They produce a report that documents all noncompliance issues. The report is peer reviewed with the SEPG. The project manager gets a chance to review the draft report for inaccuracies. When the report is finished, the results are briefed to the project team and its management chain. The project's progress toward resolving the noncompliance issues is tracked in monthly project management reviews.

When the QuEST was first formed, members of projects looked upon it less than favorably. QuEST members were thought of as process police—people who were going to get them into trouble or cause them work. We believe this was limiting our process improvement progress. We needed to change people's attitudes toward the QuEST group and process improvement in general.

QuEST decided to emphasize service to the projects. As QuEST members were doing their reviews, they tried to view all potential noncompliance issues from the project's perspective. The QuEST worked as hard on understanding the problem and making a workable recommendation as they did on finding the problem. Since most of the early problems were common across the organization, consistent recommendations led to consistent implementation of policy across the entire organization.

Once the project members perceived the SQA group as doing something for them (instead of to them), their attitudes started to change, and so did the culture of the organization. Before our Level 5 assessment, we even had a group call the QuEST and complain that nobody from their functional area had been interviewed during the last project audit. This was a complete change from the earlier attitude.

Product Lines

Sometimes, the smallest adjustment can make the biggest difference; just changing the way you look at a problem can make the solution suddenly seem obvious. For years, we struggled while we tried to make everything from 100-person projects to one-person projects fit the CMM as it was written—verbatim. This was like creating one set of rules for both football and golf. Using one model for different types of projects was the wrong thing for the wrong reason.

When we began to implement the CMM, we wanted to compartmentalize our process definition and improvement by product line. However, because we knew little about the CMM and were concerned about levels and what assessment teams might think, we decided to try to make one size fit all. It was a safe move and probably the right one at the time. However, as we began to deal with higher-maturity level issues, we realized we had to align our process improvement efforts with the way work was being done. Our early Level 4 implementation efforts failed because we were trying to force different product lines to do things exactly the same way.

One of our first Level 4 breakthroughs occurred when a few members of the Software Engineering Process Group (SEPG) visited Boeing Space Trasportation Systems, which at the time had recently been assessed at CMM Level 5. Boeing shared the concept of the ESEPG. The concept is that those who use the process should be in charge of process improvement. The visit with Boeing, plus the fact that the latest draft of CMM, Version 2 addressed the product-line concept, gave us the confidence to reorganize our process improvement initiative along product lines.

Extended Software Engineering Process Group

The ESEPG is a group at the head of each product line that takes on some SEPG duties. The group consists of project managers, supervisors, and lead engineers who work part time on process improvement issues. They work closely with the organizational SEPG to ensure efforts of each product line are consistent with organizational policy. They also coordinate defect-prevention activities at the product-line level. They meet quarterly to review defect data and causal analysis action items.

Because they are connected to the work, the ESEPGs have unique insight into the better ways to implement organizational policy and prevent defects in their product line.

By allowing the ESEPGs to implement product-line-specific solutions consistent with organizational policy, we allowed product-line differences to strengthen the organization instead of weakening it. Quantitative Process Management (QPM), Software Quality Management (SQM), and Defect Prevention (DP) activities are performed on pure product-line data by the people who manage the process.

Intergroup Coordination

When implementing the CMM, people like a discrete set of activities. Just do these tasks, and requirements management is covered; do these other activities, and project planning is satisfied. This makes for an understandable model. However, in the application of this model to real situations, the discrete independent nature of the activities is not so clear or even desirable. Do you want to have a separate meeting for every activity in the CMM that implies people need to meet? This type of implementation is advocated by some people because it is simpler to explain. It makes it easier to measure and easier to verify. And if the process were the product, it would probably be the best implementation.

However, in the real-world application of the CMM, virtually everything is integrated. Every KPA contains activities that touch or affect virtually every other KPA. The same applies to intergroup coordination. This is one of the more important KPAs, but it is mostly comprised of activities that are not activities unto themselves but are modifiers of other activities. If a requirements review is the activity, having all affected groups represented at the review is not a separate activity—it describes the right way to hold the requirements review.

The assessment team was impressed with the intergroup coordination not being just a set of activities or meetings to be attended—it was a philosophy inherent in all our activities. All groups are involved in requirements definition, all groups have input to the project plans, all affected groups are represented in peer reviews, and requirements changes are reviewed and agreed to by all affected groups. Our intergroup coordination is built directly into and distributed across the entire process.

Measurement

It seems as though the CMM says to measure everything and gives countless examples of what to measure, most of which are the kinds of things on which you would never do statistical analysis. Therefore, it is important to know the difference between measuring to know the status of an activity and measuring for statistical analysis.

For instance, minutes of a requirements review or technical percent complete from a status review are examples of measurements. From these measurements, individual managers can determine the status of their projects. However, to do statistical analysis on this data, these attributes should be measured consistently across the organization. The measurement should be made by the people doing the work and should be based on agreed-to definitions. This is why the requirement to measure is integrated throughout the CMM, but the concept of using statistical techniques is not required until Levels 4 and 5, when your measurement capabilities have grown and matured to the point where you have data that is consistent and suitable for statistical analysis.

Everything you measure costs money and takes time. To measure it with the rigor required for statistical analysis costs even more. We focused on automating the measurements we needed to produce our core metrics for cost, schedule, and quality. We measure hours of effort by process block, technical percent complete, start and stop dates, and source lines of code. For defects, we record process block injected and block detected, type, severity, and a description. Our strategy is to measure these attributes against blocks in our process.

Quantitative Project Management vs. Quantitative Process Management

Managing projects is not the same as managing processes. Project management is a real-time activity, like sailing a ship from London to New York. Once under way, your main concern is where we are now, and what is the most efficient way to get where we are going. Quantitative process management is about using data to plot a safer, faster course for the next voyage.

Unfortunately, many metrics programs are not made to do both. For instance, the Cost Schedule Control System is made to manage open, active projects—not to analyze process performance of closed projects. Many systems also are overly complicated, requiring a class in how to interpret them every time there is a management review.

Our simple metrics program treats cost, schedule, and quality as separately derived attributes of status. Cost variance is based on planned cost and incurred cost to date, which we can accurately measure. Schedule variance is based on planned schedule performance and actual schedule performance. Actual schedule performance is determined from a combination of earned value for completed milestones and technical percent complete estimates from the people working uncompleted milestones.

Cost variance is measured in dollars, and schedule variance is measured in days. Actual schedule performance is estimated as accurately as possible based on all available data—not merely completed milestones, which may be too hard to plan in small enough increments to be accurate. Our primary quality metrics are defect detection ratio (yield), defect density, and defect injection rate.

Our system works for both project management and process management. The main difference is that project management metrics are calculated on open, active projects. On the other hand, planned cost performance, planned schedule performance, budget at completion, and schedule at completion are taken from the latest negotiated agreement with the customer. Again, the emphasis is on how we get from where we are today to where we want to be.

Process management metrics are calculated from data on closed projects. The same attributes are used, but are taken from the original estimates, plus estimates from negotiated functional requirements changes. The emphasis is on making better estimates and producing more viable project plans. Averages and standard deviations for our core process management metrics and other product-line-specific metrics make up the capability baseline for each product line. These are reviewed quarterly by the ESEPG and updated with the data from projects closed within the last quarter. The capability baseline data is then used to estimate new projects.

Quantitative Process Management, Software Quality Management, and Defect Prevention

Many people think of project status in terms of cost and schedule. High-maturity organizations view status in terms of cost, schedule, *and* quality. Quantitative process management and software quality management are essentially the same activities: One is performed on cost and schedule data, the other on defect data. However, when the CMM was written, it was probably uncommon or even controversial to think of a defect detection ratio as a project status indicator.

In our implementation, quantitative process management, software quality management, process change management, and defect prevention are tightly integrated activities. One reason we jumped from a Level 3 to a Level 5 is once we had the data to do software quality management (defect types, block injected, block detected, severity, etc.), we had everything we needed to do defect prevention. Once we had the data necessary to do quantitative process management, we realized we had everything we needed to do process change management, so we pushed back our assessment approximately one year and pursued Level 5.

Our approach was to measure continuously and analyze periodically. Cost, schedule, and quality data are measured continuously by the people doing the work. We tried to automate this as much as possible, using single-point-of-entry systems. For instance, when we first started gathering defect data from peer reviews, an engineer would record anomalies on a form. Then, at the review, valid anomalies would be manually recorded in the peer review minutes. Then, the project leader would enter the defect data into a spreadsheet to make it easier to analyze. If we wanted to analyze the data from several projects together, we again somehow had to combine it. This was an overly redundant and time-consuming process.

One product line solved this problem by integrating a defect database with a CM database that was already being developed. Now, the reviewer enters anomaly data directly into a database. At the review, the anomalies are brought up online and marked as valid, invalid, or duplicate. Since it is integrated with the CM database, products cannot be configured unless all valid anomalies are resolved and checked off in the database. This database also has standard pull-down menus for defect type, process blocks, severity, and a text field to record a description. Most engineers do not mind recording data if it is quick and easy and does not get in the way of what they consider the real work. We also leveraged off our time-accounting system to measure effort by process block. Now, when people charge time to a project, they also have to attribute it to the process block.

If you minimize redundant handling or entering of the data and automate the analysis, people will participate.

The ESEPGs meet quarterly to review the data and update their capability baseline as necessary. Cost, schedule, productivity, and quality data are analyzed. Defect detection ratios and defect injection rates are calculated for each process block. Pareto diagrams are done for defect type, block injected,

Montgomery Area Golf Outing and Information Technology Partnership Day 1999

May 17 – 18, 1999

Subject: Y2K and Challenges Beyond – Government and Industry Shared Solutions

Golf Tournament and Barbecue: May 17, Wynlakes Golf and Country Club
Montgomery Area Information Technology Partnership Day: May 18, Embassy Suites Hotel, Montgomery, Ala.

Sponsored by
- Montgomery Chapter of Armed Forces Communications and Electronics Association
- Maxwell Air Force Base, Gunter Annex information technology and academy organizations
- Montgomery Area Chamber of Commerce
- Montgomery Area Chamber of Commerce Information Technology Committee
- State of Alabama, Offices of Information Technology
- Information/computer sciences departments of local universities
- City of Montgomery, Ala.

Register at
Internet: http://web1.ssg.gunter.af.mil/partnership
Voice: 334-279-5979
Fax: 334-279-5687

Chairman: William R. Stevenson
Voice: 334-416-4041
Fax: 334-416-5505
E-mail: william.stevenson@gunter.af.mil

block detected, severity, rework by process block, etc. As a result of this analysis, action items are assigned, and (if necessary) action teams are formed to further analyze process problems, causes of defects, or opportunities for improvement. The ESEPG also acts as the Configuration Control Board for product-line process changes.

Process Change Management and Technology Change Management

Because these two KPAs are similar, we combined the implementation into the same process. The Technology Change Management (TCM) process is an open process supported by an Intranet-based database accessible by everyone in the organization; anyone can submit proposals via the Intranet. The SEPG screens the proposals and facilitates as necessary. The key concept is that technology and process changes happen all the time. All we are trying to do is get people involved in using a consistent process to plan, evaluate, and implement changes. We do this by offering a way for them to formally propose ideas and resources to evaluate and test new technologies and a forum to publicize their results.

The TCM database is a repository for technology proposals and data. The SEPG only gets formally involved in the evaluation if they are requested to do so or if the proposal affects multiple product lines. Issues that affect only one product line are handled by the ESEPG. The SEPG wants to be viewed as supporting technology and process change initiatives at all levels of the division—not controlling them.

One way we evaluate major new technology or process changes is to pilot them on a small controlled scale. For example, one pilot project implemented the Team Software Process (TSP). Project members tailored a process from our existing development process, adding in the detail necessary for the TSP.

Conclusion

In the past, the term "software engineering" was usually an oxymoron. The enterprise of developing and maintaining software was not disciplined enough to be considered engineering, and the occurrence of post-release defects was considered an annoying but expected problem. Discipline is what the CMM and other such models offer. However, like any model, it needs to be applied with common sense. Treat the CMM as a guide, strive to understand the concepts on which it is based, then do the right things for the right reasons. ◆

About the Author

Patrick W. Cosgriff is an SEPG member in the Technology and Industrial Support Directorate, Software Engineering Division at the Ogden Air Logistics Center at Hill Air Force Base, Utah. Before entering the SEPG, he was the lead of the Quality Engineering Support Team. His operational background is in automatic test equipment doing test program set development and maintenance. He has a bachelor's degree in electrical engineering from South Dakota State University.

OO-ALC/TIS-3
7278 Fourth Street
Hill AFB, UT 84056-5205
Voice: 801-777-3799 DSN 777-3799
Fax: 801-775-4444
E-mail: cosgrifp@software.hill.af.mil

References

1. Paulk, Mark C., et al., *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley, Reading, Mass., 1995.